Segmentation of sidescan sonar images

Sams, Thomas; Hansen, J.L.; Thisen, E.; Stage, B.

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

Segmentation of Sidescan Sonar Images

T. Sams, J. L. Hansen, E. Thisen, B. Stage
Danish Defence Research Establishment
Ryvangs Allé 1, DK-2100 Copenhagen O

DDRE M-21/2004

Contents

1 Introduction

This report describes prototype software for sidescan sonar image segmentation developed for the Danish Naval Material Command by the DDRE.

The aim is to produce maps of seafloor types from the sidescan sonar images. These maps can then be used for planning military and mine hunting operations. The current version of the software should enable the Naval Material Command to judge whether such a tool would be of value to the Navy.

First an executive summary describing the results of the seafloor type segmentation is given. This is meant for the general reader as a non-technical self-contained description of the algorithm.

Secondly, sections describing mathematical details of the algorithm are given. Even in these sections we shall attempt to explain the algorithms in plain language, yet writing some formulae to support whom they may help.

Finally appendices describing details of mathematical methods. Further methods that were tested during the development without being applied in the final version of the program are described here.

2 Executive summary

In sidescan sonar images sand ripples are formed at short scale but constitute textures at large scale; trawl tracks form textures at large scale only, flat sand forms large scale textures from small scale characteristics, vegetation can do either of the previous. In order to make a good textural separation between regions of the images we must therefore analyse the image "simultaneously" at many scales.

Using sidescan for type-classifying seafloor is a special case of texture classification. The general texture classification problem is as yet unsolved. One of the main problems in texture segmentation is caused by the many scales involved in the problem.

In spite of these problems, we show that for some seafloor types, reasonable segmentation results can be obtained.

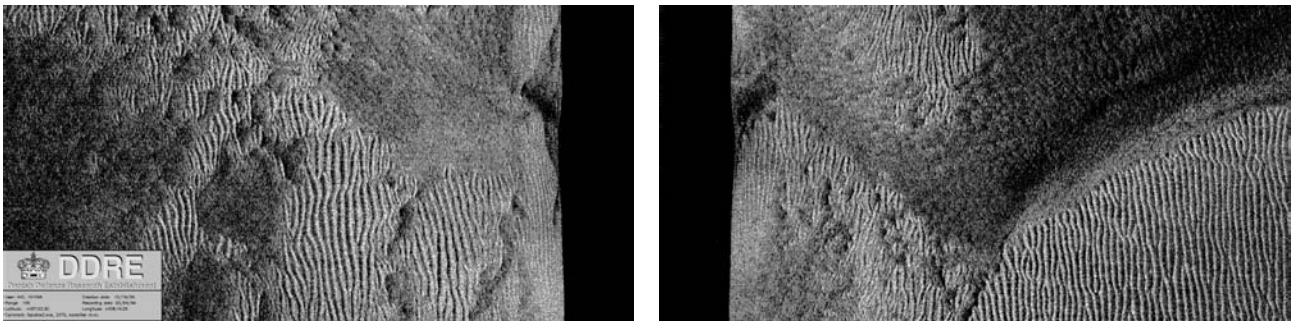

Figure 1: Sample sidescan sonar image provided by the Danish Navy shown in slant range. Actually two images, left and right relative to the tow-fish carrying the sidescan sonar, are shown. The details of the image resolution and specifics on the sonar type are not reported here, but the resolution is of order 10 cm. All images analysed in this report have this format.

2.1 Seafloor types

The initial intention was to make an algorithm that could classify any type of seafloor present around Denmark. However, this turned out to be too ambitious. First of all we have found it very hard to distinguish some types of vegetation from e. g. mud using sidescan sonar images. This is hard, even to the skilled operator: the information is probably not there in the images. (Sometimes vegetation can be identified from the shadows they cast where they meet flat sand. Otherwise it can be impossible to distinguish vegetation from mud.) Secondly, if we were to do such a segmentation, we would need reliable ground truth data, for example from video or diver inspection of the seafloor.

Instead we have focussed on producing good results on a few types of seafloor that we can be sure of: Flat sand, sand ripples, scattered stones, and trawl tracks. Types that are not identified in the current version of the software are: vegetation, mud, and gravel: they all end up in the "unknown" type. Examples of the different seafloor types are shown in figure 2 and examples containing elements of unknown type are shown in figure 3.

2.1.1 Flat sand bed

Flat seafloor simply consisting of plain sand grains. Not very soft. Characteristics: Uniform gray, almost no variation in the picture. Occurs when there is only very little water current or when the currents are too high for sand ripples to form.

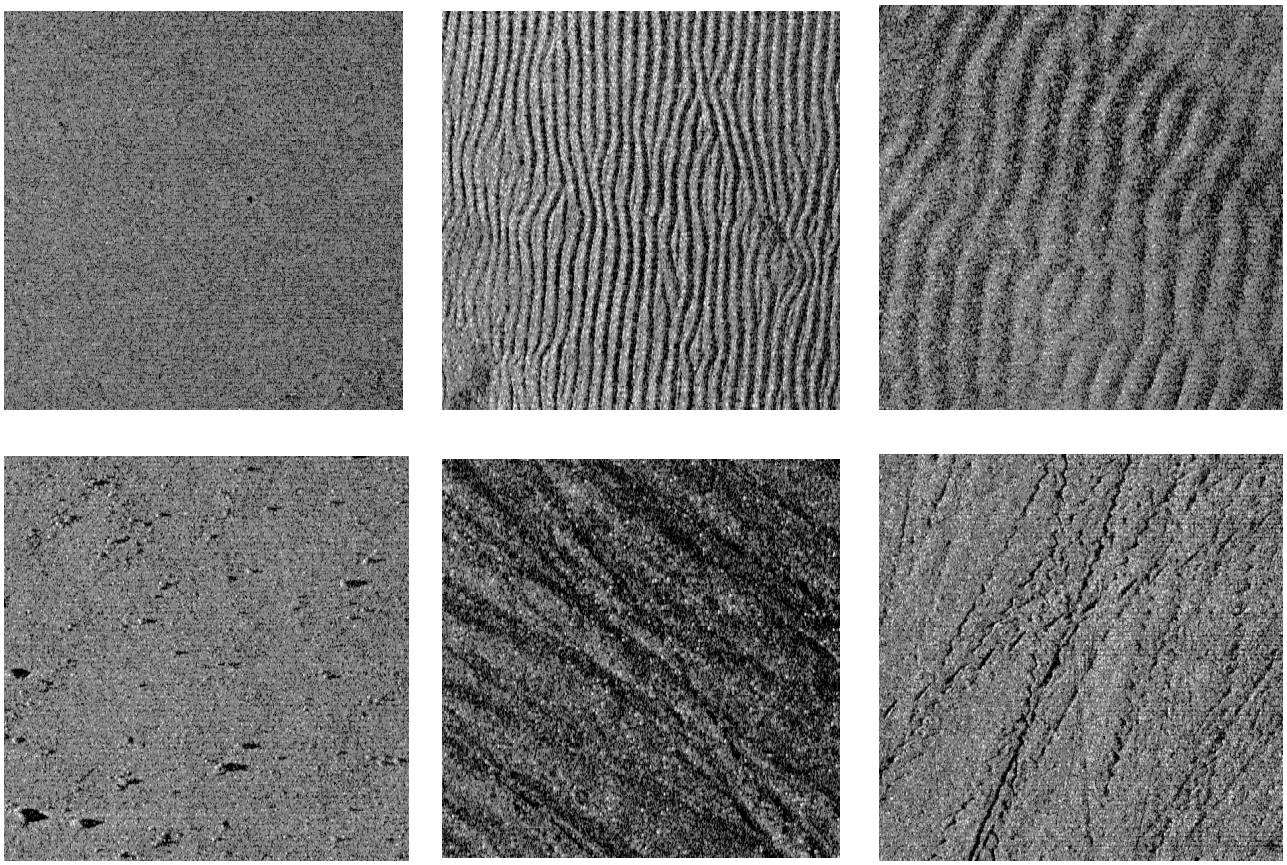

Figure 2: Examples of the different seafloor types classified. Top left to bottom right: *Flat Sand Bed, Short Ripples, Long Ripples, Scattered Stones* and *two different kinds of trawl tracks.*

2.1.2 Sand ripples

Sand ripples occur where the seafloor consists of freely moving sand grains and as the water flows at intermediate, possibly oscillatory, currents. Characteristics: parallel ripples perpendicular to the current. Wavelength from a few cm to many meters.

2.1.3 Scattered stones

"Scatterede stones" are defined as flat sand with small and intermediate sized scattered stones. The objects are recognised as consisting of a highlight and a shadow zone behind it. However, the highlight is not always present. Characteristics: stone sizes vary, but we look for stones typically of the order 10×10 pixels. The density of stones required for the region to be considered as scattered stones is of order 1.5 in a region of 100×100 pixels.

2.1.4 Trawl tracks

In large regions the seafloor around Denmark is covered with trawl tracks. If currents are low, they can stay there for months or even years. They are characterised by long almost straight lines on an otherwise flat seafloor. If they are on soft sand/mud the background can be quite dark. If on larger-grain sand the background is uniform gray. In either case, a newly formed trawl track shows up as a dark straight line.

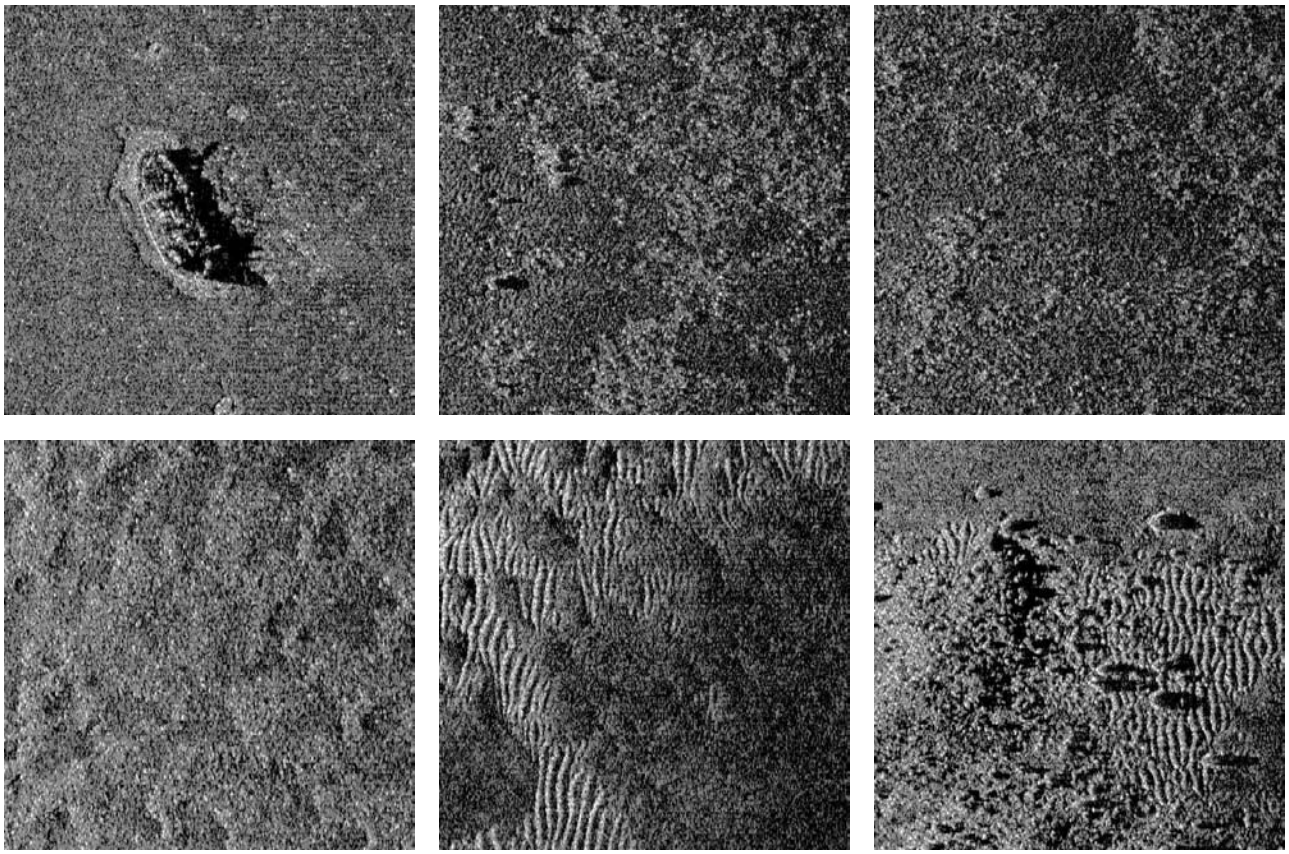

Figure 3: Examples of sea floor containing regions which cannot directly be classified into one of the categories flat sand, sand ripples, scattered stones, or trawl tracks.

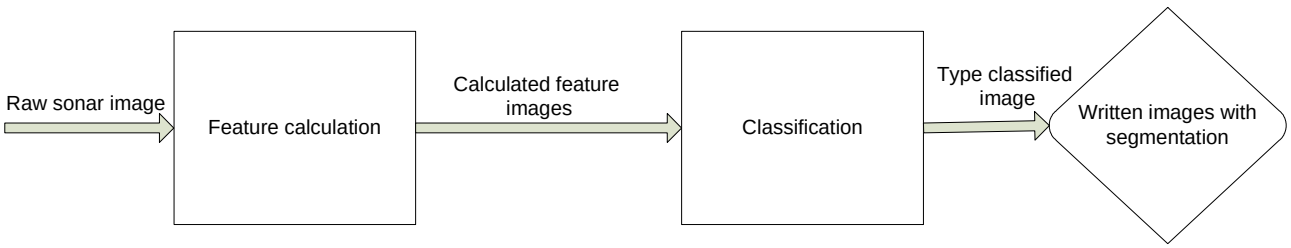

Figure 4: Flowchart of the algorithm.

2.1.5 Unknown type

If an area cannot be classified into one of the types described above, we denote it as unknown. This is due to the fact, that we have not yet been able to develop methods that with satisfactory confidence detects them. Types of seafloor falling into this category include e.g., mud, seaweed, gravel and larger shadows. Characteristics: Dark areas. Often containing irregular patterns.

2.2 Algorithm

There are two major steps in our classification algorithm.

Feature calculation: The first processing of the sidescan sonar images is the feature calculation. These features include e.g. mean or variance of gray scale values in a area around a pixel. The result of the feature calculation is feature vectors, one vector per pixel of the original image.

Feature classification: Sequentially we then ask whether a feature (or a few features) provide evidence for a pixel belonging to a particular type of seafloor.

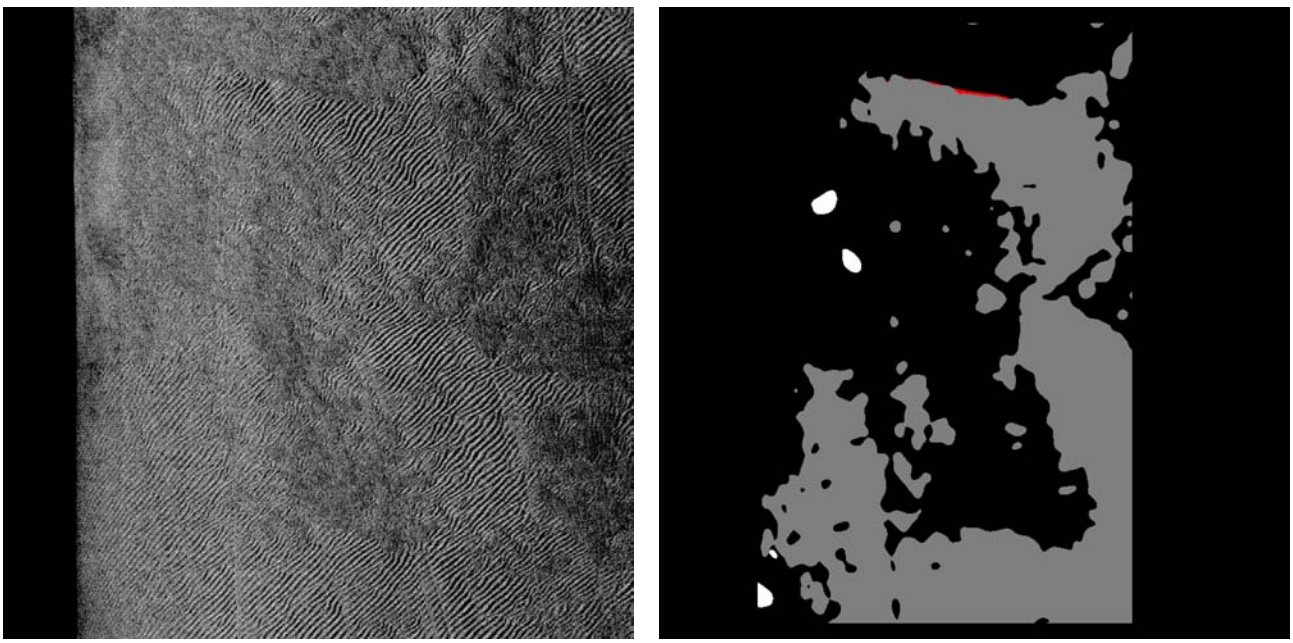

Figure 5: Left: Example of sidescan sonar image with ripples. Right: Classified image with ripples (gray).

Our classification scheme is based on local features calculated from sonar images. The features are calculated as a statistics in the neighbourhood of a point in the image. The size of neighbourhood varies, and in the case of the trawl track detector the neighbourhood it is quite large.

Our approach is to calculate dedicated features designed to detect specifically the types of seafloor present around Denmark. The seafloor types for which we have been able to make good detectors for are those shown in figure 2, i. e. flat sand, short sand ripples, long sand ripples, scattered stones, and trawl tracks. Below we briefly describe the detectors for each of these types, and show examples of segmentation. A more technical description of the features can

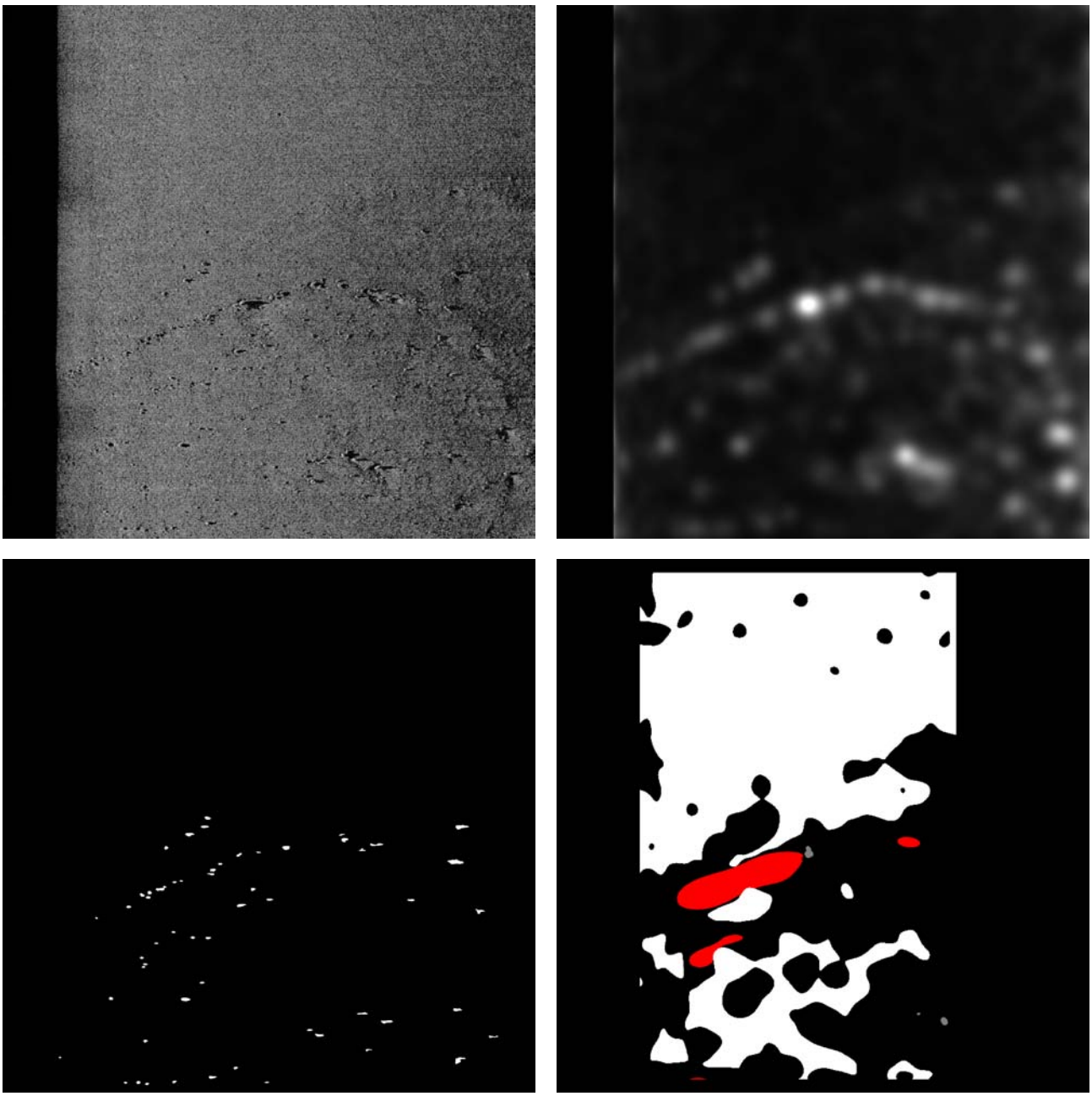

Figure 6: TL: Example of sidescan sonar image with flat sand and some scattered stones. TR: Local variance of image. BL: Detected shadows of stones. BR: Classified image with flat sand (white), scattered stones (red), and unknown (black).

be found in section 3.1. (A different approach, where features are picked from a large set of wavelet coefficients, has been developed by Grasso and Spina [3].)

The segmentation which follows the feature calculation is performed using simple thresholding on most of the features. In fact, a 2-d thresholding in a combination of gray-level and type-specific features is used for all features but the trawl tracks. This is a particularly simple version of more general cluster analysis which shall be discussed in appendix D.

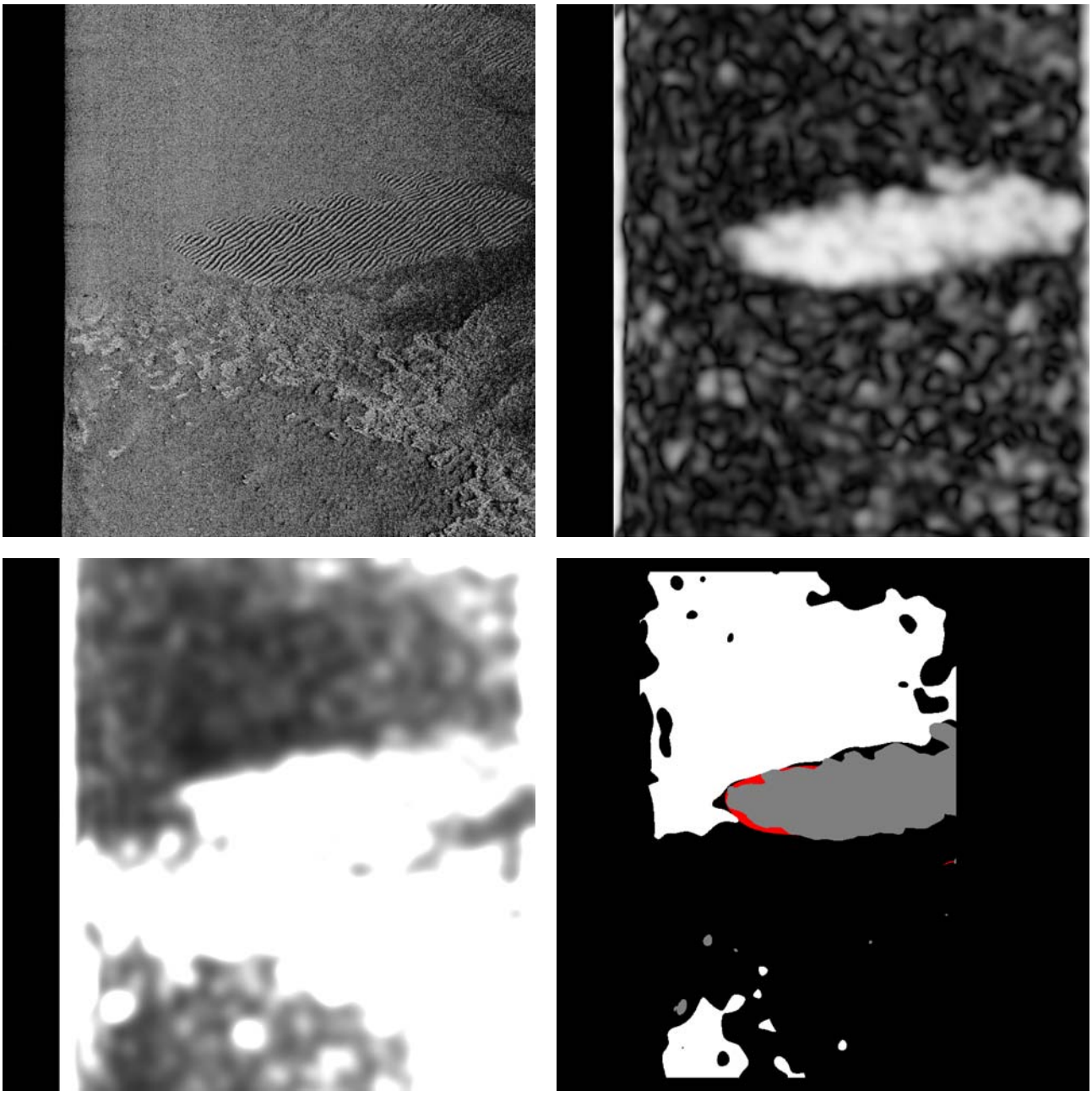

Figure 7: TL: Example of sidescan sonar image with flat sand and a small region with sand ripples. TR: Short wavelength ripple detector output. BL: Flat seafloor detector, i. e. local variance of image. BR: Classified image with flat sand (white), ripples (gray), and unknown (black). At the edge of the sand ripples erroneously detected scattered stones (red) are seen.

2.2.1 Flat sand detector

Flat sand is detected by the simplest detector of all: a threshold on the local variance of the image in combination with a threshold on the local average gray value. In figure 6 a sidescan sonar image with some flat sand regions is shown.

2.2.2 Sand ripple detection

Sand ripples are detected using a dedicated symmetry-detector. This detector measures the local symmetry of the slopes of the gray-level. If the slopes locally are strongly symmetric the

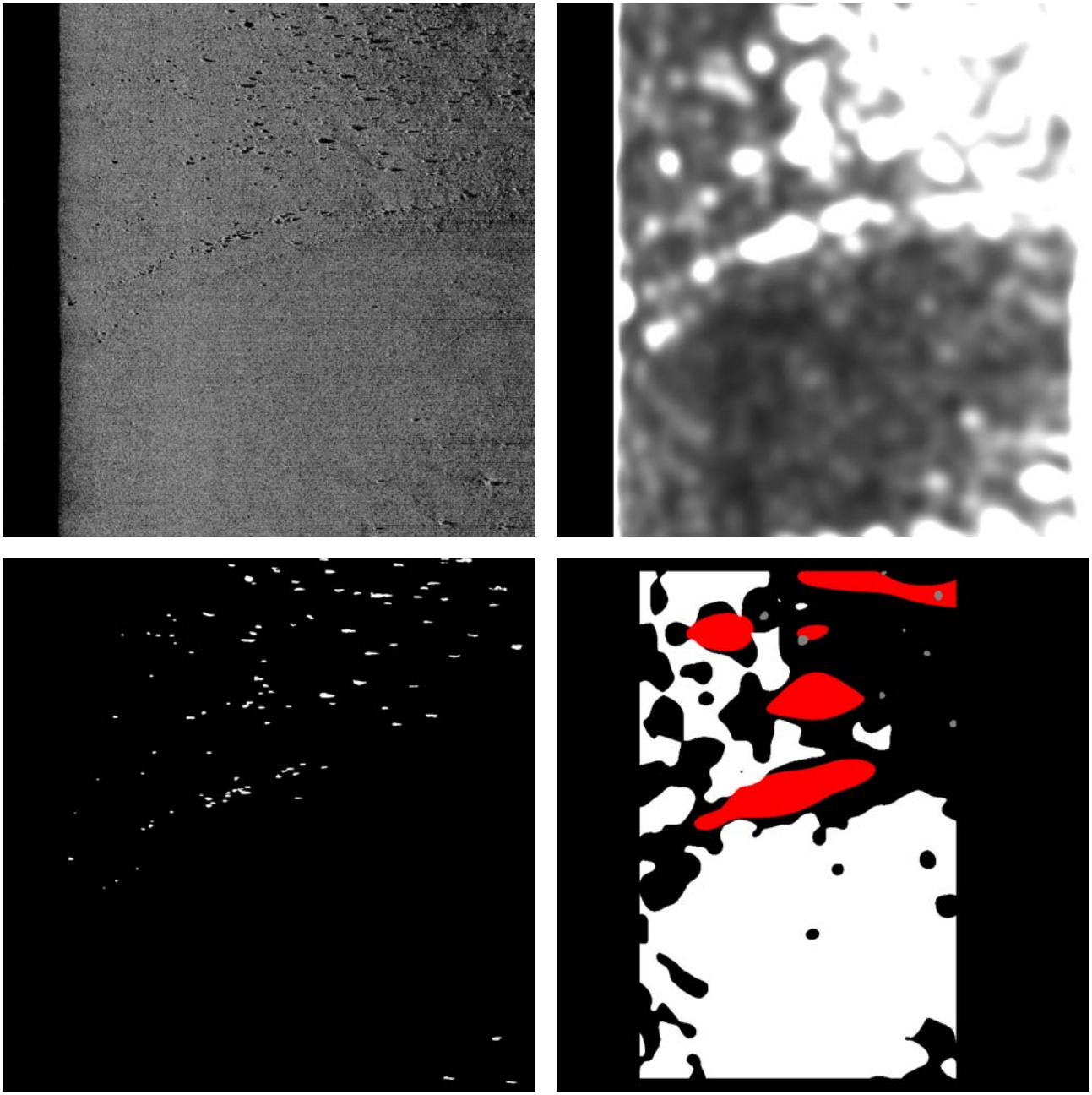

Figure 8: TL: Example of sidescan sonar image with flat sand and some scattered stones. TR: Local variance of image. BL: Detected stones. BR: Classified image with flat sand (white), scattered stones (red), and unknown (black).

sand ripple type is assigned. Examples of detected ripples are shown in figures 5 and 7.

2.2.3 Scattered stones detection

Stones are detected using using only the shadow they cast. First an algorithm enhancing connected dark regions is applied to the image. This image is then thresholded and searched for connected regions. All connected shadow regions within a specified size are then assumed to be cast by stones. Finally the density of stones is calculated locally. If the resulting density is above a threshold the type "scattered stones" is assigned. An example with scattered stones and flat sand is shown in figure 6.

2.2.4 Trawl track detection

Trawl tracks are identified in the sidescan sonar images as long dark straight lines. When both long lines are sufficiently dark and a specified fraction of shorter lines constituting the long lines are dark, the trawl track type is assigned to a band around the trawl track. An example with detected trawl tracks is shown in figure 10.

2.3 Classifier

Once the features are calculated at all pixels, the seafloor is classified using a simple threshold classifier. Two sets of thresholds are specified for each type: an allowed range for the specific feature detector, and an allowed range for the locally averaged gray level.

The classification is done in a specific order: 1) trawl track with high confidence, 2) long wavelength sand ripples, 3) intermediate wavelength sand ripples, 4) short range sand ripples, 5) flat sand, 6) scattered stones, 7) trawl tracks with lower confidence. This means that the

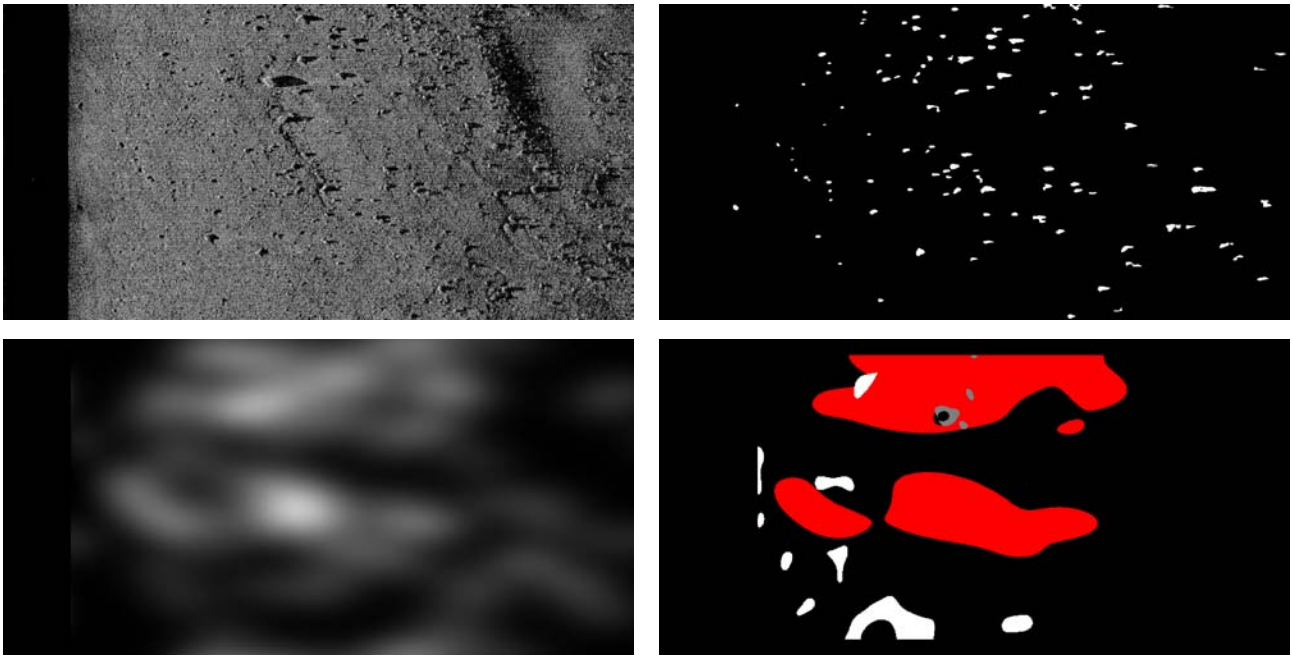

Figure 9: TL: Example of sidescan sonar image with scattered stones. TR: Detected shadows of stones. BL: Local density of stones. BR: Classified image with scattered stone region marked red. (The white regions have been detected as flat sand. The gray regions have erroneously been detected as sand ripples.)

former have priority over the latter in the classification reflecting the ranking of confidence in the type assignments.

2.4 Overview maps

The individual classified small sections of sidescan sonar images are not of much help to the operator who may be in the process of planning a minesweeping operation. He would rather be requesting an overview at a much larger scale. An example of such a large-scale overview is shown in figure 11.

It is our feeling that such overviews could be of interest to the military planner. The aim of the present report is therefore to provide the reader with the impression, that reasonable quality overview maps are within reach. Yet, not give him the impression that all problems are solved.

2.5 Discussion and relation to other work

The approach to seafloor segmentation presented in this report differs from those found in the literature in that we use dedicated detectors selected by hand for each seafloor type. This has the disadvantage of not being adaptive. On the other hand we have been able to make good detectors of e. g. scattered stones and trawl tracks. Both types are important around Denmark, the former because stones often are minelike, the latter because trawl tracks cover a large fraction of the seafloor.

Both scattered stones and trawl tracks would be very hard to grasp within a framework of conventional image analysis tools like Fourier analysis, wavelet analysis, or statistical distribution analysis (see e. g. Grasso and Spina [3]).

Probably, a combined approach could be fruitful.

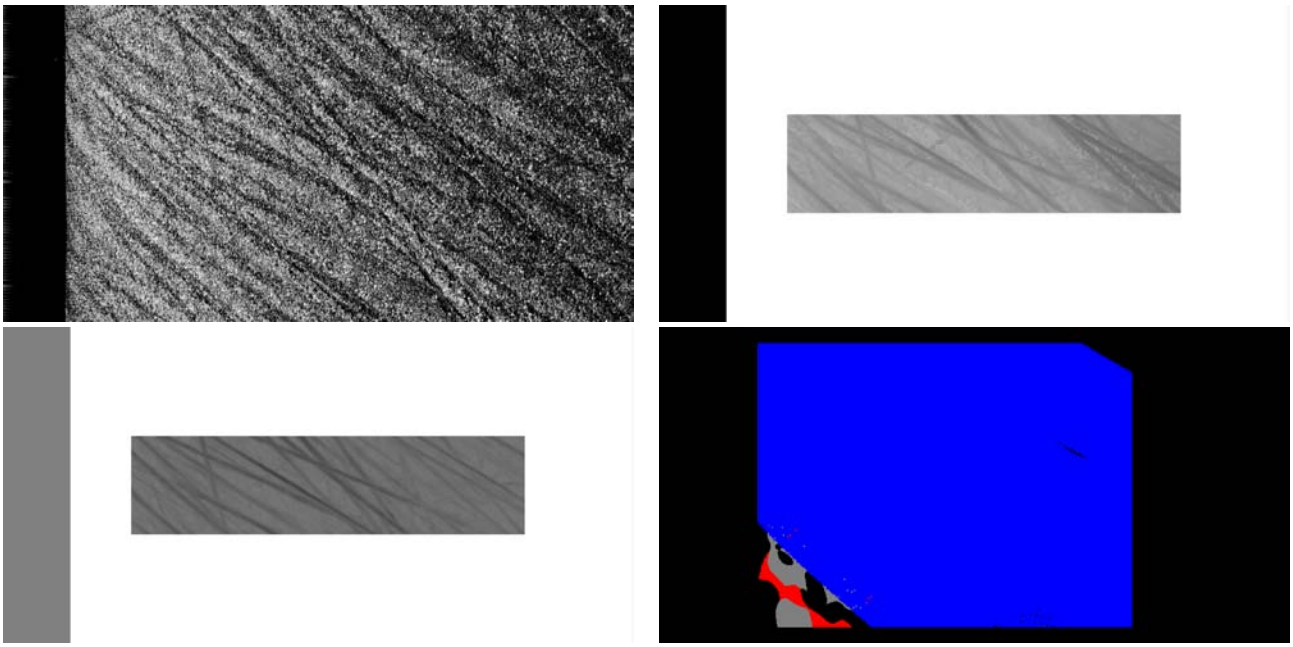

Figure 10: TL: Example of sidescan sonar image with trawl tracks. TR: Raw, minimised "Radon" integrals. BL: Background subtracted "Radon" integrals. BR: Classified image with trawl regions marked blue.

2.6 Recommendations

We suggest that the Danish Navy Material Command, SMK, consider seafloor classification algorithms for inclusion in a more general sidescan sonar analysis and presentation tool. The present report gives a flavour of what can be done along the directions of automatic classification of large areas of seafloor: At the 50-100 m scale very useful maps of seafloor types can be generated. At scale below 10 m there are significant uncertainties in the sidescan sonar image segmentation.

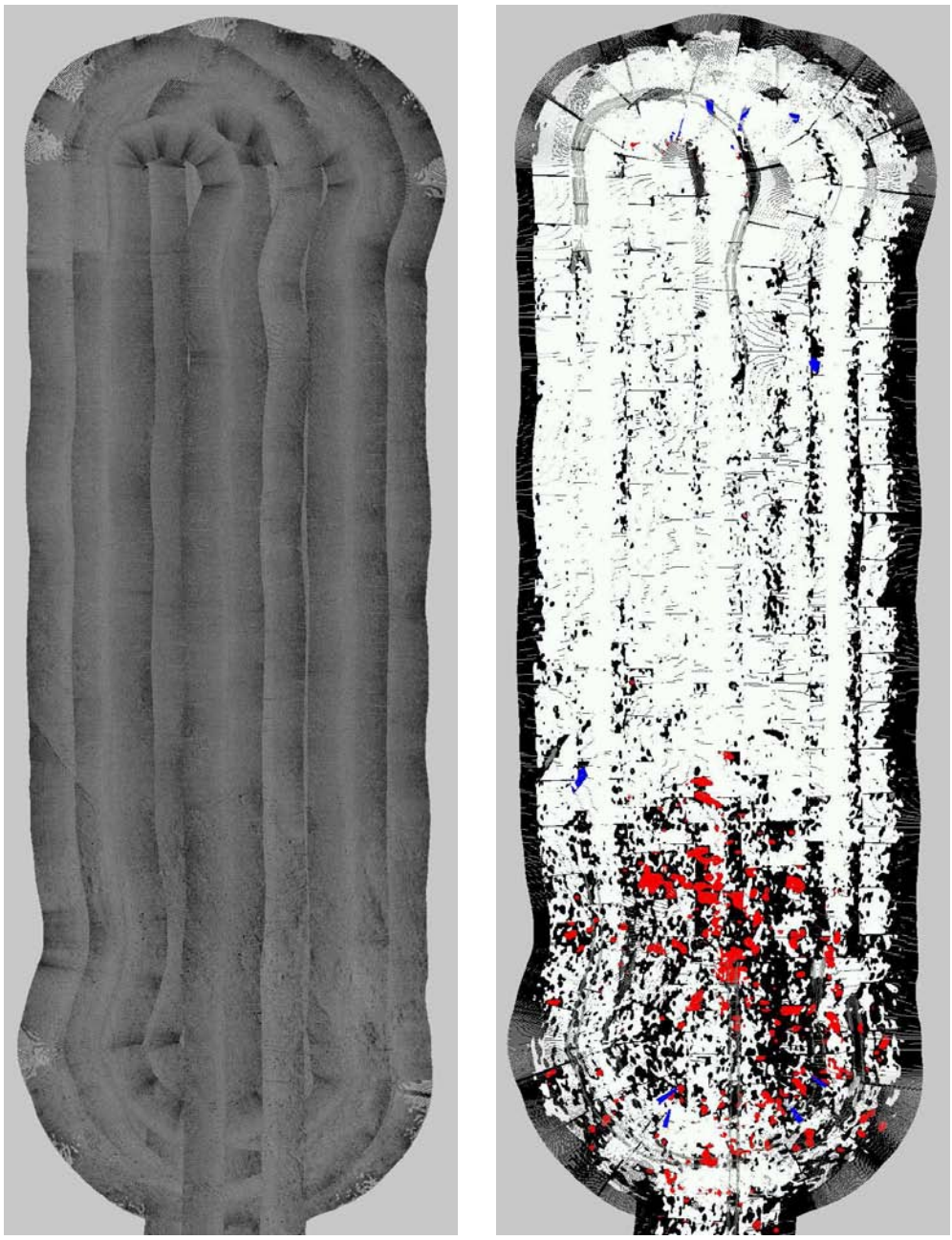

Figure 11: Example of large segmented area. The seafloor classifier does a reasonable job in identifying regions with flat sand (white) and scattered stones (red).

3 Details of algorithm

3.1 Feature selection

A crucial point in the seafloor classification is the selection of a proper set of features. Several issues have to be taken into consideration. First, we want to have as few different features as possible, since the number of calculations needed for a classification grows with the number of features. Further, introducing redundant or irrelevant features introduces both statistical and systematic noise to the decision making. Secondly, we need to have enough features to be able to see a difference between the different seafloor types. As a rule of thumb we need a number of features equal to the number of seafloor types that we are interested in.

The feature selection is actually the most important step in the whole process. Finding good features for our special purpose involved close examination of the data. It turned out, that for each of the seafloor types under consideration, we were able to find a feature, that could distinguish that seafloor type from the rest.

All features are calculated setting options in the program `feature.cpp`. A listing of the options are presented in appendix E.1.

3.1.1 Flat sand detection

Flat sand detection is performed by gating the mean gray value and thresholding the standard deviation. More specifically: First the image $u(x, y)$ is pre-smoothed by folding with a Gaussian kernel to ignore fluctuation below scale σ

$$v = \langle u \rangle_\sigma = K_\sigma * u. \tag{1}$$

Then the mean value at scale ρ is calculated

$$u_{\text{mean}} = \langle v \rangle_\rho. \tag{2}$$

(Really, this is just the average at a slightly larger scale $\sqrt{\sigma^2 + \rho^2}$.) Finally the fluctuation of the pre-smoothed image is calculated at scale ρ

$$f_{\text{flat}} = (\langle (v - u_{\text{mean}})^2 \rangle_\rho)^{\frac{1}{2}}. \tag{3}$$

This used as the primary detector of flat sand. Figure 12:TL shows that this is reasonable, since f_{flat} separates flat well from other seafloor types. The selection of the presmoothing scale σ and the fluctuation scale ρ was done such as to optimize the separation between flat sand and other seafloor types.

3.1.2 Scattered stone detection

A single stone, large enough to be detected, is seen in a sidescan sonar image as a black shadow. Sometimes, but not always, there is also a highlight coming from the bit of the stone pointing perpendicular to sonar beam. The slant-range width w_{slant} of the shadow can be translated into a height of the stone using

$$h = \frac{d}{s}\, w_{\text{slant}} \tag{4}$$

where d is the distance from the sonar to the seafloor and s is the current slant range. The along-track dimension of the shadow may be directly interpreted as the size of the stone along that direction.

In order to make a clear detection of the individual stones we first enhance the dark regions over the lighter using a moment-filter

$$u_{\text{shade}} = (\langle(\langle u\rangle_\sigma)^m\rangle_\rho)^{1/m}. \tag{5}$$

In order to enhance the lower gray-values a moment of order less than unity is used. A few experiments led us to chose $m = 0.2$. In words: the image is pre-smoothed at scale σ, then lifted to power m, and finally smoothed at scale ρ. As with all other features the scale selection contained in choosing the parameters σ and ρ is crucial to the performance of the filter.

The image u_{shade} obtained using equation 5 is then thresholded resulting in an image of shadows. Finally the height and along-path size of each stone is thresholded, and the centre of mass of each valid stone is kept. If the density of valid stones is above a second threshold, the region is assigned the type "scattered stones".

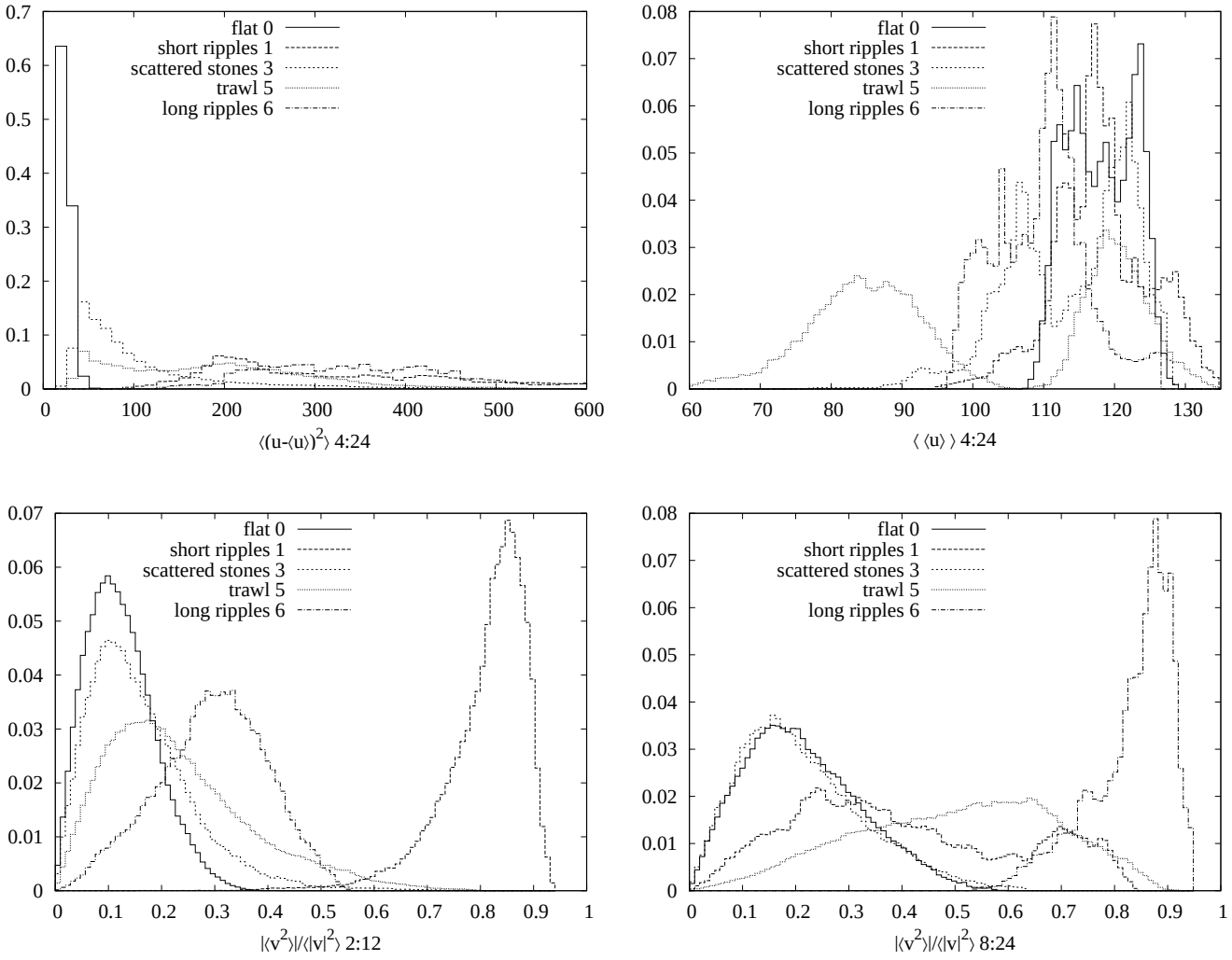

Figure 12: TL: Distribution of local variance of gray value with scale parameters $\sigma = 4$ and $\rho = 24$. (Flat sand detector.) TR: Distribution of local average at scale 24. This feature is included in order to avoid very dark regions typically from mud or larger shadow areas. Note that the trawl data has two peaks coming from trawl tracks on silt (dark) and sand (lighter) respectively. BL: Distribution of sand ripple detector Σ_2 with scale parameters $\sigma = 2$ and $\rho = 12$. BR: Distribution of sand ripple detector Σ_2 with scale parameters $\sigma = 8$ and $\rho = 24$.

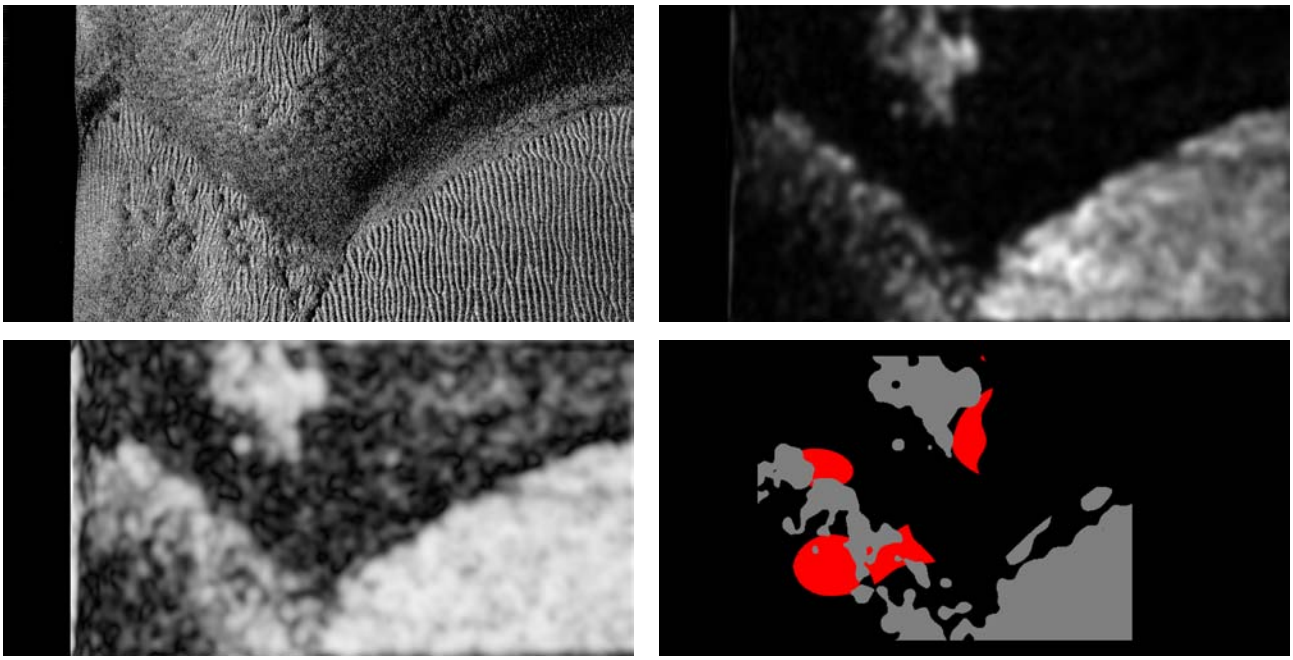

Figure 13: TL: Sidescan sonar image from the North Sea with sand ripples and some mud/vegetation. TR: The gradient energy Λ_0 with $\sigma = 2$ and $\rho = 12$ applied to the sidescan image. BL: The ripple detector Σ_2 with $\sigma = 2$ and $\rho = 12$. BR: Classified image with sand ripples (gray) and erroneously detected as scattered stones (red).

3.1.3 Sand ripple detection

Even though sand ripples are characterised by a large degree of periodicity we have found that Fourier methods are not the most efficient way of detecting them. Instead we have developed a dedicated filter which is described in detail in references [1, 6]. The detector measures the symmetry of the gradient field of the sand ripples

$$\vec{v} = \nabla \langle u \rangle_\sigma. \tag{6}$$

The gradient vector of the pre-smoothed field is then written in the complex plane as

$$\boldsymbol{v} = \nabla \langle u \rangle_\sigma = v_x + \mathrm{i}\, v_y = v \mathrm{e}^{\mathrm{i}\varphi}. \tag{7}$$

We calculate the gradient in the x-direction using the optimised 3×3 filter [7]

$$\begin{pmatrix} -3/16 & 0 & 3/16 \\ -10/16 & 0 & 10/16 \\ -3/16 & 0 & 3/16 \end{pmatrix} \tag{8}$$

and similarly for the y-component.

By coupling $\boldsymbol{v}$ to itself we may construct objects which are invariant under a rotation of π. $\boldsymbol{v}\overline{\boldsymbol{v}}$ and $\boldsymbol{v}\boldsymbol{v}$ span the full space of such vector couplings. (Here $\overline{\boldsymbol{v}}$ is the complex conjugate of $\boldsymbol{v}$.) These objects are now averaged locally to form measures of symmetry

$$\Lambda_0 = \langle \boldsymbol{v}\overline{\boldsymbol{v}} \rangle_\rho = \langle v^2 \rangle_\rho \tag{9}$$

and

$$\boldsymbol{\Lambda}_2 = \Lambda_2 \, \mathrm{e}^{\mathrm{i}2\varphi_2} = \langle \boldsymbol{v}\boldsymbol{v} \rangle_\rho = \langle v^2 \mathrm{e}^{\mathrm{i}2\varphi} \rangle_\rho. \tag{10}$$

In order to avoid an oscillating symmetry signal, the scale ρ should be larger than the scale of the structures of interest. The dominant orientation is then given by the angle φ_2. Note that the phase of $\mathbf{\Lambda}_2$ is two times φ_2.

A natural normalised measure of the strength of the 2nd order symmetry is then the ratio

$$\Sigma_2 = \frac{\Lambda_2}{\Lambda_0} \tag{11}$$

with values in the interval $[0; 1]$. This is identical to the orientation measure given by Bigun *et al* [2].

The gradient energy Λ_0 may be seen as a measure of strength of the gradient field, while Σ_2 measures how much of the strength comes from line structures.

In figure 13 we show an example of the gradient energy Λ_0 and the 2nd order detector Σ_2 applied to a sidescan sonar image of seafloor partially covered with sand ripples. In combination, the detectors show remarkable ability to separate sand ripples from other textures in the image.

3.1.4 Trawl track detection

Trawl tracks are seen in the sidescan sonar images as broad dark lines. In ground coordinates the line is typically straight on a 100 m scale.

In order to detect the long dark lines we calculate the integrals along lines in all directions passing through a pixel. Apart from the fact that we only integrate up to a finite length, this is similar to a Radon transform. In order to speed up the calculation of the line integrals passing through a point we use a method inspired by Jang *et al* [4]: First short integrals are calculated in low angular resolution. Then the length is doubled and the angular resolution likewise, and the new integrals are calculated as a sum of two half-length integrals.

Once the integrals in all pixels at the desired angular resolution are calculated, background subtracted integrals can be calculated by subtracting the integrals parallel to the current one.

The minimal background subtracted integral is now found and thresholded. If below threshold, the non-background-subtracted integral at the same angle is also thresholded.

Finally, a user-specified fraction of the original-length background-subtracted integrals at the same angle are required to satisfy the threshold.

An example of the resulting classification is shown in figure 10.

3.2 Classification

After the calculation of features, classes are assigned using a simple threshold classifier. All but the trawl type are assigned using the type detector in combination with a thresholding on the local average. As mentioned earlier the classification is performed in a specific order

1. trawl track with high confidence,
2. long wavelength sand ripples,
3. intermediate wavelength sand ripples,
4. short range sand ripples,
5. flat sand,
6. scattered stones,

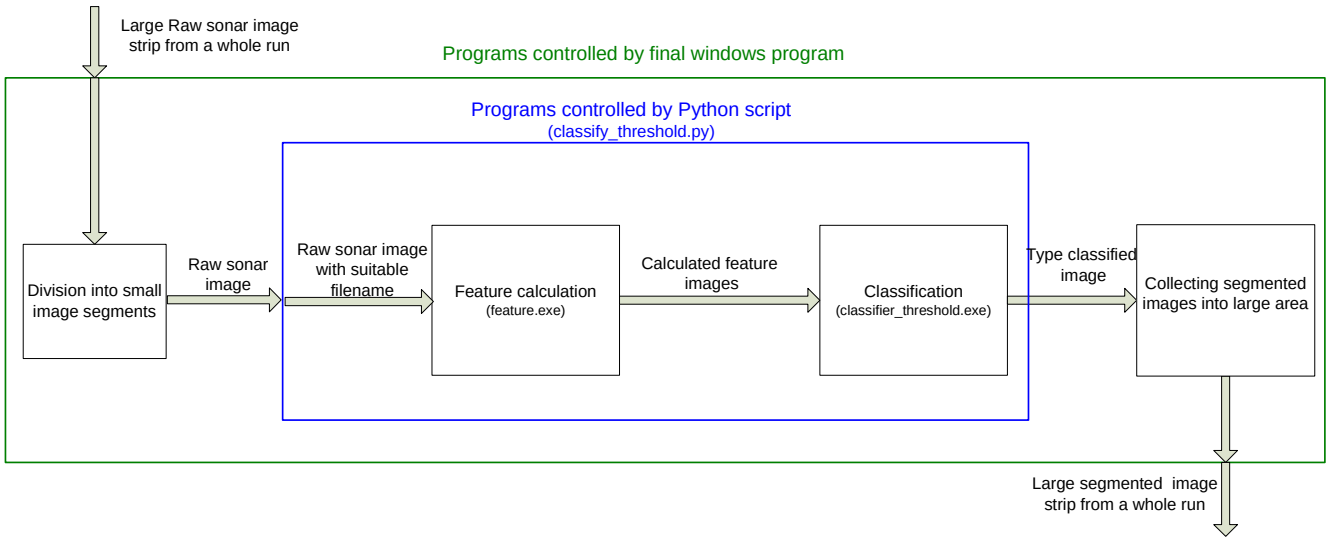

Figure 14: Flowchart of the implementations.

7. trawl tracks with lower confidence.

All thresholds are set conservatively. This means that quite large regions of "unknown" will be assigned, even in regions where the human eye would be able to classify.

The thresholding is done using the program `classifier_threshold.cpp`. The options of this program are listed in appendix E.2.

3.3 Implementation

The algorithms are implemented as a series of several small programs with limited functionality, and clearly defined input and output functions written in C++. The small programs are then glued together by an advanced interpreted script language, Python. All parameters used in running the compiled C++ programs are set in the Python script file. A pseudo-flowchart for the implementation is shown in figure 14.

3.4 Algorithm precision

We have examined the results of the algorithm run on a large sonar data set. From the examination the following successes and issues with the algorithm are noted:

- Sand ripple detection is very precise. Almost all areas with sand ripples are found, and there are almost no false detections.
- Sometimes, when there is a sharp boundary between two types of seafloor, the algorithm detects the area as being trawl, or scattered stones.
- A single large stone can sometimes be detected as a single sand ripple.
- Scattered stones are sometimes assigned in regions with many dark spots. This can for example occur in vegetation or in muddy areas.

Appendices

A The sonar data

The sidescan sonar images are all obtained with the Danish Navy Sidescan sonars, which are run by the Standard Flex 300 mine sweepers. The sonar is positioned on a "fish" which is towed behind an unmanned vehicle sailing in approximately a straight line, then turning 180^0, and sailing back, with a small overlap with the first part of the route. In that way a larger area is swept, and there is a considerable overlap in the sonar data, which helps avoiding noise. The images obtained from the sonar are relatively easy to inspect by the naked eye. By a closer inspection it turns out, that the data has several flaws, which make them rather difficult to examine by algorithms.

A precision issue is, that the data are only described by 4 bits corresponding to only 16 gray-tones per pixel. Further, there is no description given of the way the measured values decrease with distance from the sonar. Clearly, some correction is done on the data, but the form is not easily found by inspection of the data. Other problems with the data are described in the following.

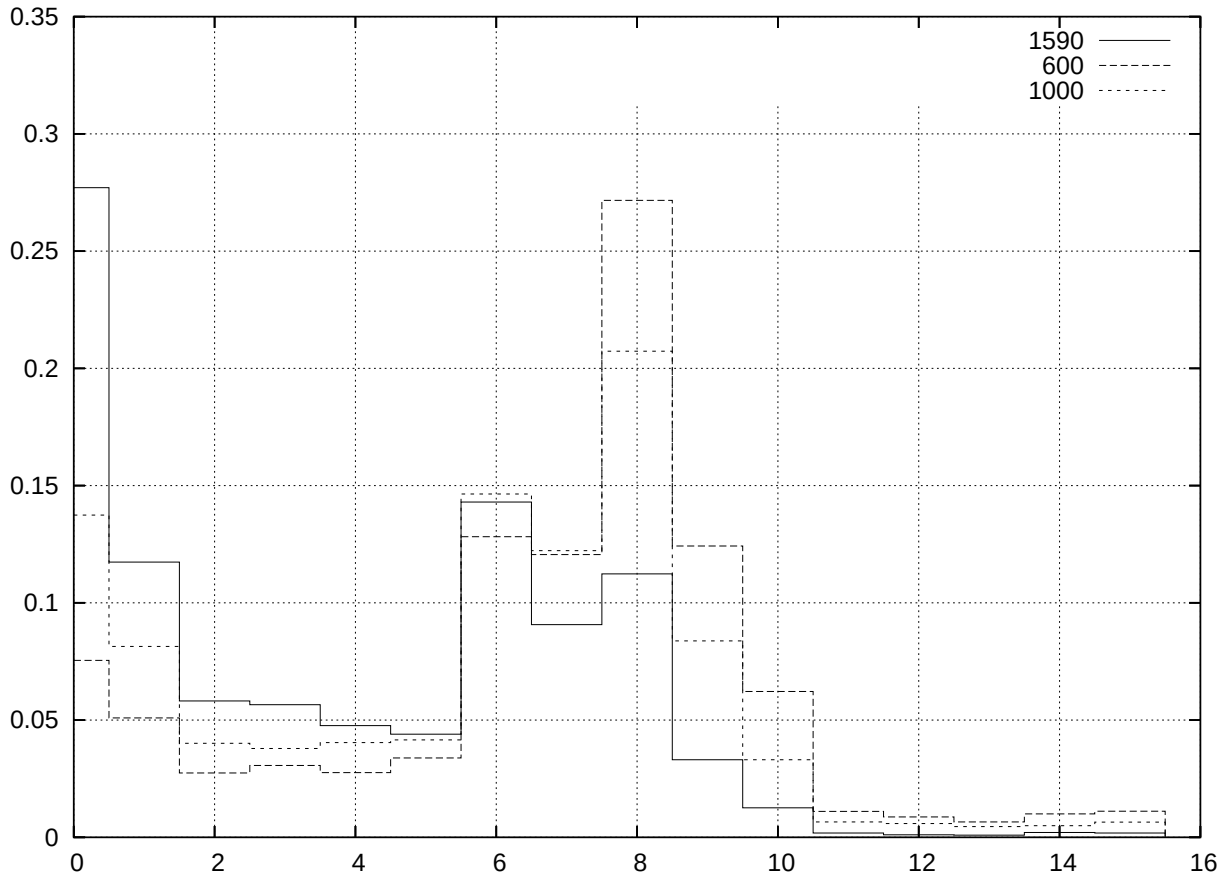

Figure 15: Histogram of the values present in a very large amount of data. Notice the non-uniformness of the histogram - apparently every second value is less likely to occur compared to its neighbours. The histograms show statistics for data in pixel positions 1590 (full line), 1000 (dotted line) and 600 (dashed line) from the sonar

A.1 Value uncertainty

Unfortunately the data presented to us from the Danish Navy Sidescan sonar has been preprocessed. Probably this preprocessing serves to optimise the 4-bit (16 gray tones) images for viewing. This however has the disadvantage, that processing the data by computer is made harder, since we are not dealing with the raw data. We have not been able to acquire knowledge of the exact form of the function, that has been applied to the data.

By analysing a very large amount of data and making histograms of the 16 possible values in the data, it turns out, that some of the 16 values are preferred to others, and the less preferred values are not just the max and min values. An example of such a histogram can be seen in figure (15). This calculation has been done on data from several of the navy sonars and the result is the same.

A.2 Spatial uncertainty

Occasionally the fish "wobbles" a little while it is dragged forward. The x-direction from scan to scan are therefore not exactly parallel. The sidescan sonar, holding an array of 8 hydrophones, scans for eight positions in the y-direction simultaneously. Because of this, there is a tendency in the data to have a "period-8" structure, meaning, that for every 8^{th} line in the sonar images, there is a discontinuity in the intensities, which does not reflect anything physical on the seafloor. This means that the gradient is quite noisy when passing from one 8-block to the next. It reality, the period 8 does to a large extent reduce the resolution of the data in the sailing direction with a factor of 8. We have tried several methods to exclude the effect of this periodic noise. We have found it possible develop a method to remove the periodic noise in a way so the periodical structure is not as apparent when the images are examined by the naked eye. However, the noise is still disturbing the calculations performed on the images, e. g. the gradient. We have therefore chosen not to use any form of correction of this periodic noise.

B Preprocessing, filtering

All features are calculated using at least one preprocessing: a presmoothing of the image at a scale usually called σ. The scale is unique to the feature in question.

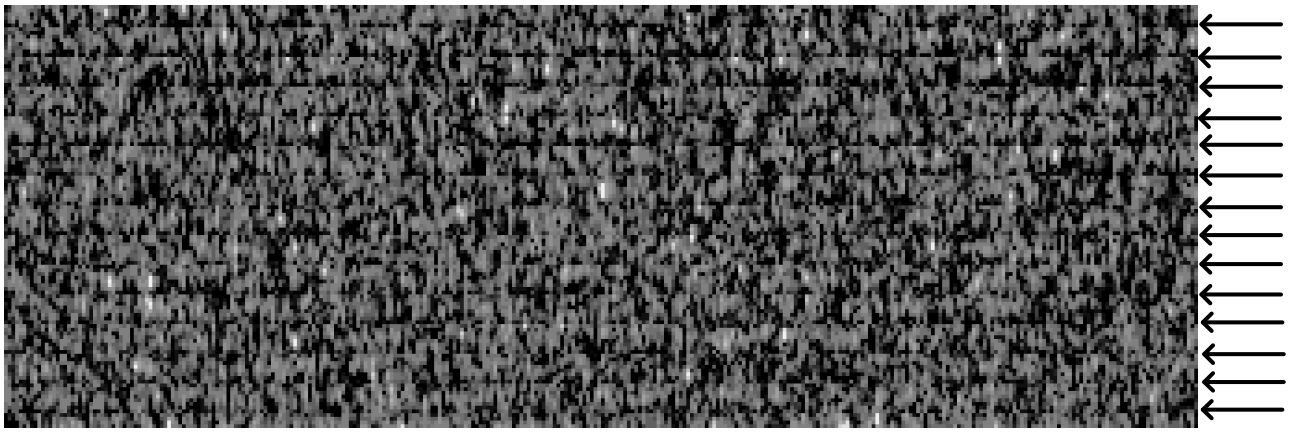

Figure 16: Close-up of a sonar data recorded over a flat sand bed. Both the noise in the data as well as the periodical nature of the data is apparent

Further a distance correction of the intensity of the slant range intensity of those features that are sensitive to this is performed. In particular, for the trawl track features and the mean value this is of importance.

We did some testing on the use of median filtering for the trawl track an scattered stones features. However, we were not able to get any improvements out of those attempts. Thus they are not included in the final version of the algorithm.

C Examined features

Several possible candidates for features for the classification algorithm have been examined in this work. Even though, only a few of them are used in the final product, we think that some of the other features are worth mentioning. A few of these features perform at the same quality level as those used in the final algorithm. The simpler is then preferred over the more complex.

C.1 Jacobian characteristics

If the (sonar) image is regarded as a function depending on the image coordinates, it is possible to calculate the Jacobian matrix $\boldsymbol{J}(x, y)$, in each point (x, y) of the image. The Jacobian is a 2×2-matrix holding the three possible second derivatives of the image. The eigenvalues and eigenvectors tells something about the local curvature of the image values. Sand ripples, e. g., holds one large eigenvalue, with the corresponding eigenvector being perpendicular to the ripples, while the smallest eigenvalue is close to zero. The Jacobian features turned out to perform just as well, but no better, than the symmetry measures. Being simpler, the latter were then chosen to be used in the actual algorithm.

C.2 Unser features

Apart from mean and variance, also higher moments, inverse moments and features inspired by statistical mechanics (entropy, energy, ...) can be calculated. One such set of features is described by Unser [9]. We have tested them. But they do not lead to improved classification compared to what we already can do.

D Examined classification methods

The set of features constitutes a vector space. This multi dimensional space is then examined for clusters. When the classification is supervised, volumes corresponding to specific types are given a priori (from the training set) in the feature space. It is then a matter of testing whether a given pixel is positioned inside one of the volumes, and if this is the case, that pixel is designated the type corresponding to the volume in question. There are several ways to do this volume division in feature space listed in the literature. Apart from the simple thresholding of subsets of features used in the final version of the algorithm, we have tested a covariance method, or maximum likelihood method and a simple thresholding method. Finally, a cluster analysis algorithm called “isodata” has been given consideration for inclusion in the algorithm.

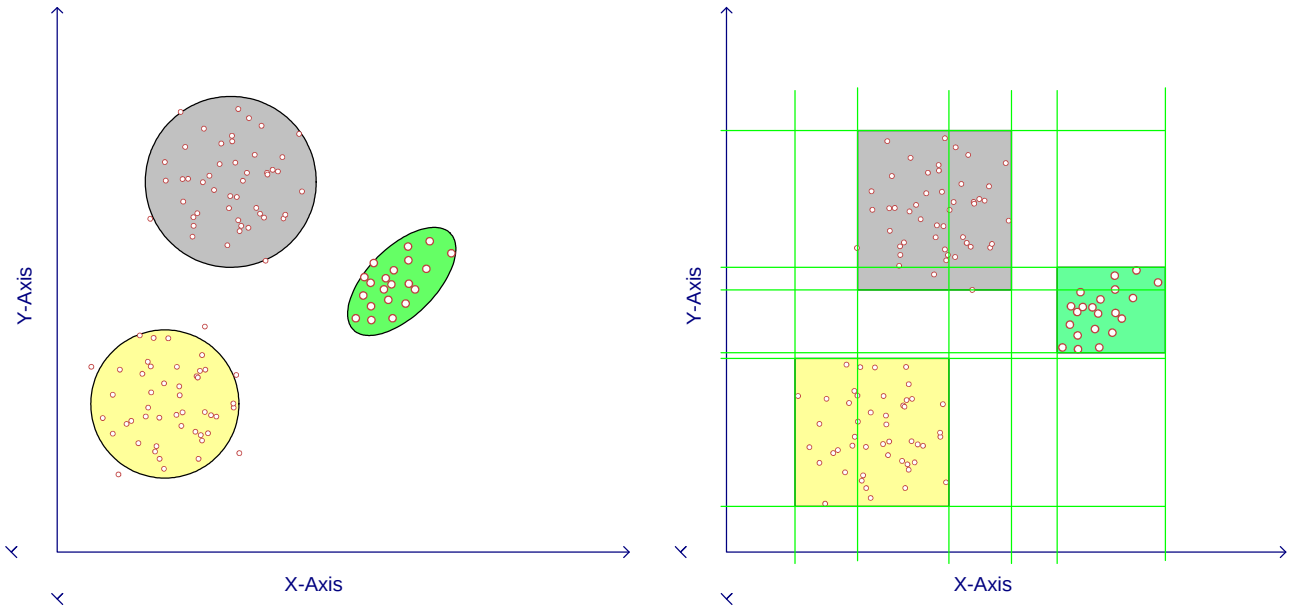

Figure 17: The principles of the covariance method (left) and the thresholding method (right). The ellipses show the area of interest for the covariance method in the left figure, while lines marking the threshold values for each type for the thresholding method are seen in the right figure.

.

D.1 Covariance method

This method is based on the calculation of the distance in feature space from the mean of the manually selected training set for each seafloor type. The covariance matrix is used to determine how many standard deviations a given point in feature space is from the centre of the distribution in feature space. The principles behind the division of feature space is seen from figure 17 left.

Let Ω_j be the training set for type j. Given a feature vector $\boldsymbol{f}$, the mean $\boldsymbol{m}_j$ and covariance matrix $\boldsymbol{\sigma}_j$ for a specific type j are then calculated as

$$\boldsymbol{m}_j = \langle \boldsymbol{f} \rangle_{\Omega_j} \tag{12}$$

$$\boldsymbol{\sigma}_j = \langle (\boldsymbol{f} - \boldsymbol{m}_j)(\boldsymbol{f} - \boldsymbol{m}_j)^T \rangle_{\Omega_j}. \tag{13}$$

The $\langle \cdot \rangle_{\Omega_j}$ denotes averaging over the training set Ω_j for seafloor type j. For a given point $\boldsymbol{f}$ in feature space to be classified, the distance to the type j in units of standard deviations is then given as

$$S_j = (\boldsymbol{f} - \boldsymbol{m}_j)^T \boldsymbol{\sigma}_j^{-1} (\boldsymbol{f} - \boldsymbol{m}_j) \tag{14}$$

for all types using the corresponding covariance matrices and mean positions. The point is then classified as belonging to the class with the smallest distance.

Note that one should calculate the statistical significance of the hypothesis of compatibility with a given type. If neither of the types are accepted, "unknown" should be assigned. Using this procedure, one may actually use different feature vectors for each seafloor type.

D.2 Thresholds

A conceptually simple method, based on giving an upper and a lower threshold for each type in each feature dimension in the feature-vector space. The principle is easily seen from figure

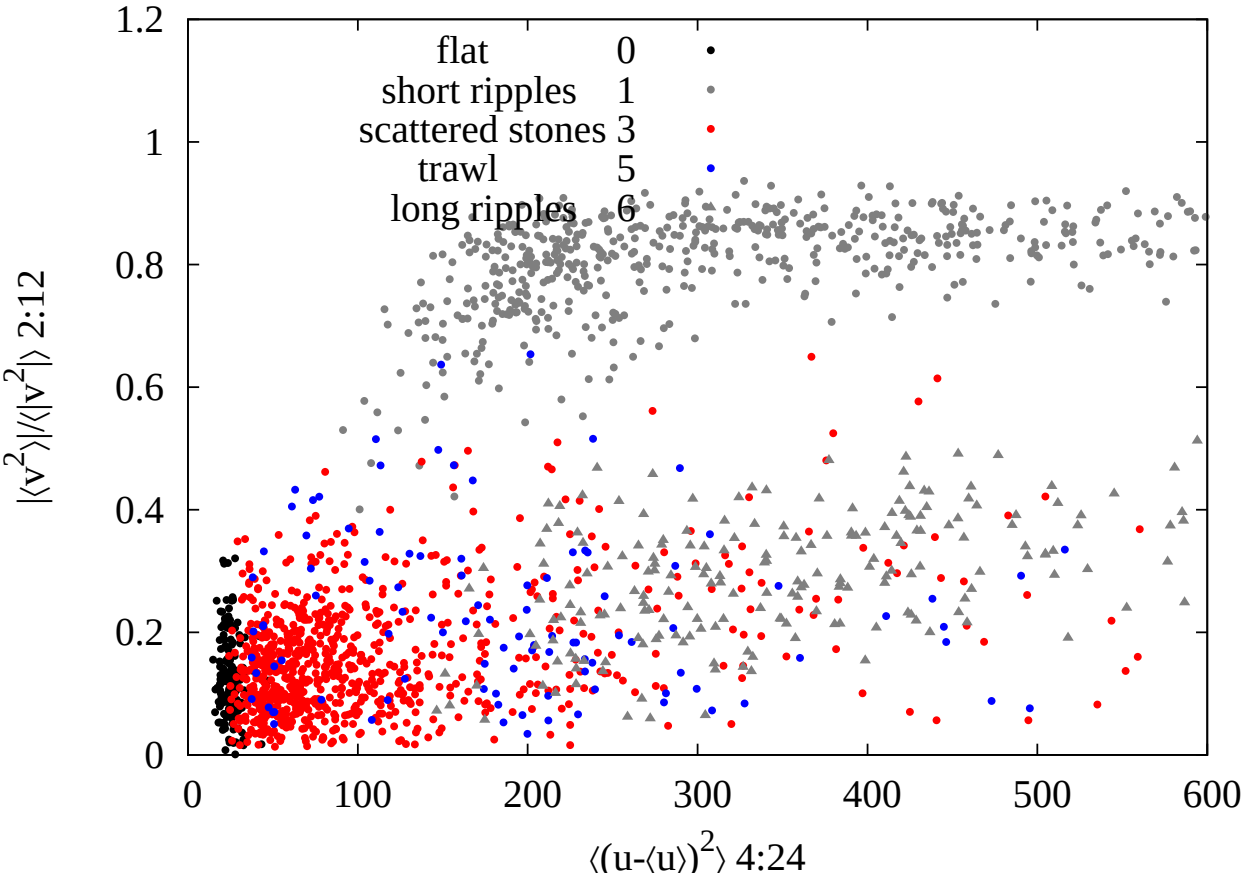

Figure 18: 2-d scatterplot of short ripple detector, $\sigma = 2$, $\rho = 12$, versus the fluctuation in the image on scale $\rho = 24$.

17 right. In figure 18 a scatterplot of the short ripple detector output versus the variance is plotted for the different seafloor types.

D.3 Classification quality check

One of the problems when using a supervised classification algorithm is the selection of suitable training sets in feature space. These training sets have to be clearly enough in feature space for the classification algorithm to be able to uniquely distinguish between the classes. If there is an overlap in feature space between the different training classes one may select supplementary features which allow a better separation.

D.3.1 Jeffreys-Matusita distance

One method, implemented in the DDRE product for this test is calculation of the so called "Jeffreys-Matusita distance", J [5, 8]. J is a measure of how well two densities, $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$, in a multi dimensional space are separated. Here $\boldsymbol{x}$ is a point in the multidimensional space.

$$J = \int_X \left[\sqrt{p_1(x)} - \sqrt{p_2(x)}\right]^2 \mathrm{dx} \tag{15}$$

This can also be written as

$$J = 2(1 - p), \tag{16}$$

where p is given as

$$p = \int_X \sqrt{p_1(x)p_2(x)}\ \mathrm{dx}. \tag{17}$$

The centre of the density $p_i(x)$ is U_i and the covariance matrix of the density is Σ_i. If $p_1(x)$ and $p_2(x)$ are multivariate Gaussian densities, then J can be written

$$J = 2(1 - e^{-\alpha}), \tag{18}$$

where

$$\alpha = \frac{1}{8}(\mathrm{U}_1 - \mathrm{U}_2)^{\mathrm{T}}\Sigma^{-1}(\mathrm{U}_1 - \mathrm{U}_2) + \frac{1}{2}\log_e\left[\frac{\det\Sigma}{\det\Sigma_1 \cdot \det\Sigma_2}\right] \tag{19}$$

and

$$\Sigma = \frac{1}{2}\left[\Sigma_1 + \Sigma_2\right]. \tag{20}$$

J ranges from 0 to 2, where 2 is infinite separation and 0 are coinciding densities. These equations are easily implemented and used as quality testing of the separation of the training classes in feature space.

D.3.2 Using isodata for quality checking of supervised classification

ISODATA is an unsupervised algorithm used to automatically divide a set of points positioned in an n-dimensional space into clusters. The algorithm uses an initial guess on cluster as seed for the algorithm, and then, based on given threshold for maximum distance between clusters and the maximum allowed variance of the points in each cluster, it divides or merges clusters in each iteration, until a steady state is reached. We have done several studies involving the isodata algorithm. We have not tested the algorithm thoroughly enough to include the isodata algorithm in the first version of the DDRE algorithm.

We propose a new method to test whether the types for a supervised training set are separated enough in feature space. The principle behind the algorithm is to first select the proper training set, and calculate the corresponding points in feature space. Then isodata is used on the set of all these training values for the different types. If the class division, that isodata comes up with of this set, corresponds to the supervised types, the supervised data set is probably uniquely determining classes. If there is a large discrepancy between the training classes, and the isodata classes, the training sets for the different types in feature space, are probably overlapping.

E Programs and their parameters

E.1 feature.cpp

```
feature.exe [flags] <filename>
  -o arg  output filename (without extension), output --> fno.##.tif
  -f arg  number of first output file (0 is also a number!)

  -F dy:dx[:sigma]  preceed feature calculation by median filter
          median filter using box of size dy*dx, where dy and dx are odd
          if sigma>0 preceeded by Gaussian presmothing

  -b w:m  Background subtraction to minimise x-dependence of gray level
          w <int>    width of background subtraction area
          m <float>  maximal background subtraction
          Note: Background subtraction is done after downsampling. Thus
          the width should be given in downsampled coordinates.


  -D arg  dy:dx[:save]
          dy:dx   downsampling factors. Note that all features are
                  calculated using the downsampled image. Thus the scales
                  for the feature flags are implicitly multiplied by these
                  factors, i.e. feature -d1 -e4:12 and feature -d2 -e2:6
                  are scale-equivalent.
          save:   1 => save downsampled images feature.dwn.##.tif (default)
                  0 => don't


  -G depth[:flag]
          slant --> ground range conversion
          depth <int> depth from surface to seafloor
                       0 => depth is estimated
                      >0 => depth is value read
                      depth should be given in original coordinates,
                      typical value: -G170
          flag <int>  0 => slant-->ground NOT performed, but value of depth
                       is read/estimated (to be used in -T flag)


  -e s:r[:#,...,#]
          s:r scales for linear diffusion of image and exp(i*m*phi)
              0      <|v|^2>
              1      |<v*|v|>|/<|v|^2>
              2      |<v^2>|/<|v|^2>
              3      |<v^3/|v|>|/<|v|^2>
              4      |<v^4/|v|^2>|/<|v|^2>
              5      |<v^5/|v|^3>|/<|v|^2>
```

```
            6       |<v^6/|v|^4>|/<|v|^2>
            7       |<v*|v|>
            8       |<v^2>|
            9       |<v^3/|v|>|
           10       |<v^4/|v|^2>|
           11       |<v^5/|v|^3>|
           12       |<v^6/|v|^4>|


-j s:r  Jacobi features:
        s   scale for presmoothing of image
        r   scale for diffusion of features tr(J), det(J), and tr(J)^2


-m s:r[:#,...,#]
        s:r scale for presmoothing and averaging in calculation of variation
            0       aver   <<u>_s>_r
            1       vari   <u_s^2>_r - <u_s>_r^2


-M s:r[:#,...,#]
        s:r scales for linear diffusion of moments
            0       <(v-min)^1>^(1/1)
            0       <(max-v)^1>^(1/1)
            .
            .
            9       <(v-min)^10>^(1/10)
            9       <(max-v)^10>^(1/10)


-r s:r  scales for presmoothing of image and diffusion of structure tensor


-p s    do a Gaussian preesmoothing at scale <s> before specific feature
        algorithm


-R r:m:M:w:d:[t0,t1]:[t2,t3]:f
   r       int   resolution
   m       int   initial integration length, must be power of 2
   M       int   final integration length, must be power of 2
   w       int   width of integration band, must be an odd number
   d       int   distance between foreground and background bands
   t       float thresholds
   [t0:t1] float bounds on raw line integrals
   [t2:t3] float bounds on backgroundsubtracted line integrals
   f       float fraction of initial length pieces that must satisfy
           threshold separately
```

```
    Calculates integral of length maxlinelength starting at minlinelength
    multiplying the length in each step by 2
    The angular rsolution is set such that the outermost point is within
    the specified resolution.
    Typical values: -v -G170 -D4 -b100:20 -R1:8:128:3:4:[-1000,-.035]:[0,.34]:.75 -f13
Calculates 4 features:
 0 Minimal backgroundsubtracted integral along line of length M.
 1 Minimal integral without background subtraction
 2 [t0,t1] && [t2,t3] thresholded version of 0 and 1
   1 added if [t0,t1] OK, 2 added if [t2,t3] OK, 4 added if fraction of
   initial pieces satisfying threshold satisfies threshold
 3 Number of lines through point satisfying all criteria larger than f,
   i.e. 1+2+4=7.


 -T s:r:M:t:f:[L0:L1]:[H0:H1]:g:S      Threshold in moment + counting of objects
         s <float>  presmoothing scale
         r <float>  postsmoothing scale
         M <float>  moment (e.g. 0.2)
         t <float>  threshold in moment to identify regions
         f <int>    0 => regions where below threshold
                    1 => regions where above threshold
         L0 <float> minimal length of object (shadow/highlight)
         L1 <float> maximal -      -  -       -
         H0 <int>   minimal height an object (shadow/highlight)
         H1 <int>   maximal -      -  -       -
         g <float>  scale for counting objects
         S <int>    1 => create shadow shifted image, i.e. image where
                    shadow length is contracted to just-below-sonar and
                    shifted to the expected position of the corresponding
                    highlight
Calculates 3 or 4 features:
0  Moment: <(<u>_s/(max-min))^M>_r^(1/M)
   measures tail of distribution
1  <threshold(S0,Moment,S1)>_g
   calculates the density of regions of size within boundaries
2  Simple threshold image of first image
3  Shadow shifted image
Typical set of parameters:
feature -v -g170 -d2 -D -b200:20 -T1:2:0.2:0.28:0:5:1000:20:1 -f6 feature.tif
feature -v -g170 -d2 -D -b200:20 -T0:1:5.0:0.65:1:3:1000:20 -f10 feature.tif


 -u s:r[:#,...,#]              selected Unser features
         s:r scale for presmoothing and averaging
         do Unser features with Gauss aver at scale r
             0      aver   mean(y,x)
             1      vari   (u(y,x) - mean(y,x))**2
```

```
            2,3      corr   (u(y,xp) - mean(y,x)) * (u(y,xm) - mean(y,x))
            4,5      cont   (u(y,xp) - u(y,xm))**2
            6,7      homo   1/(1+(u(y,xp)-u(y,xm))**2)
            8,9      shad   (u(y,xp) + u(y,xm) - 2*mean(y,x))**3
           10,11     prom   (u(y,xp) + u(y,xm) - 2*mean(y,x))**4

  -v      increase verbosity level by 1

          Input is read from filename.tif.
```

E.2 classifier_threshold.cpp

```
classifier_threshold.exe -d dymin:dymax:dxmin:dxmax -T [T0,T1]:[T2,T3]...
   -t <type> -z <feature.##.tif> [feature.##.tif ...] <file.best.tif>

Simple threshold classifier.
Applies thresholds (given by -T flag) to the a set of (feature) image
files. If all features are within their corresponding thresholds, the
class pixel in the feature.best.tif is set to the value given by the
-t flag.

Input:
 -d arg             dxmin, dxmax minimal distances from boundary of image
                    dymin, dymax minimal distances from boundary of image

 -T arg             thresholds between which signal must be to be assigned
                    the type given by the -t arg. [T0,T1] is applied to the
                    first feature.##.tif file, [T2,T3] to the next, ...

 -t type            type number to be assigned if thresholds given in -T
                    arg are met. Default: 255.

 -z                 reset <file.best.tif> to 255 before new assignments

 <feature.##.tif>   feature in which to set threshold
 <file.best.tif>    best type file 0,1,...,255. Only points in != 255 in
                    this file are assigned a new value.

Output:
<file>.best.tif     (modified) best type file
<file>.classified.tif  color version of <file>.best.tif using colors in
                       atppalette.h
```

E.3 classify_threshold.py script

Since all parameters for the programs `feature.exe` and `classifier_threshold.exe` are controlled by the a python script we simply give a listing of what it does here. Given an input

file `feature.tif`, first a number of features are calculated and placed in files `feature.##.tif`. Then the classification is performed using the features files as input. The output of the classification is placed in `feature.class.tif` and `feature.classified.tif`.

```
feature -v -G170 -D2 -e 1:6:2 -f0  feature.tif
feature -v -G170 -D2 -e 2:9:2 -f1  feature.tif
feature -v -G170 -D2 -e 4:12:2 -f2  feature.tif
feature -v -G170 -D2 -b200:20 -u 2:12:0,1 -f4  feature.tif,
feature -v -G170 -D4 -b100:20 -R1:8:128:3:4:[-2,-.05]:[0,.29]:0.875
        -f13 feature.tif
feature -v -G170 -D4 -b100:20 -R1:8:128:1:4:[-2,-.07]:[0,.32]:0.9375
        -f23 feature.tif
feature -v -G170:0 -D1:2 -b200:20 -T0.5:1.5:0.2:0.20:0:[2,20]:[1.5,10]:40
        -f17 feature.tif

# Simple threshold classification:
# simple classification into
# 0  white       flat sand         {255,255,255} v
# 1  gray        short ripples     {127,127,127} v
# 2              gravel            {  0,255,255}
# 3  dark gray   scattered stones  {255,  0,  0} v
# 4  green       vegetation        {  0,255,  0}
# 5  blue        trawl             {  0,  0,255} v
# 6  gray        long ripples      {127,127,127} v
# 255            not classified    {  0,  0,  0} v

classifier_threshold -d40:40:250:400 -T[0.5,10000]
  -z -t5 feature.16.tif feature.best.tif
classifier_threshold -d40:40:250:400 -T[.8,1]:[80,135]
  -t6 feature.02.tif feature.04.tif feature.best.tif
classifier_threshold -d40:40:250:400 -T[.6,1]:[80,135]
  -t1 feature.00.tif feature.04.tif feature.best.tif
classifier_threshold -d40:40:250:400 -T[.7,1]:[80,135]
  -t1 feature.01.tif feature.04.tif feature.best.tif
classifier_threshold -d40:40:250:400 -T[100,135]:[0,42]
  -t0 feature.04.tif feature.05.tif feature.best.tif
classifier_threshold -d40:40:250:400 -T[85,130]:[.00030,.020]
  -t3 feature.04.tif feature.18.tif feature.best.tif
classifier_threshold -d40:40:250:400 -T[0.5,10000]
  -t5 feature.26.tif feature.best.tif
```

References

[1] Finn Thomas Agerkvist and Thomas Sams. *Symmetry detectors in 2-D images.* DDRE report F-19/2003, 2003.

[2] Josef Bigün, H. Granlund Gösta, and Johan Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transitions on Pattern Analysis and Machine Intelligence*, 13:775, 1991.

[3] Raffaele Grasso and Francesco Spina. *Unsupervised sea bottom classification from sidescan sonar images using multi-resolution transform features.* SACLANT UNDERSEA RESEARCH CENTRE REPORT SR-372, 2003.

[4] Jeong-Hun Jang and Ki-Sang Hong. Fast line segment grouping method for finding globally more favorable line segments. *Pattern Recognition*, 35:2235–2247, 2002.

[5] H. Jeffreys. *Theory of Probability.* Oxford, Oxford University Press., 1948.

[6] Thomas Sams and Finn Thomas Agerkvist. *Coherence enhancing diffusion filtering of sidescan sonar images.* DDRE report F-38/2002, 2002.

[7] H Scharr, S Körkel, and B Jähne. Numerische isotropieoptimierung von fir-filtern mittels qerglättung. In E. Paulus and F. M. Wahl, editors, *Munstererkennung 97*, pages 367–374, Braunschweig, 1997. Springer.

[8] P.H. Swain and R. C. King. Two effective feature selection criteria for multispectral remote sensing. *LARS Technical Note*, 042673, 1973.

[9] M. Unser. Sum and difference histograms for texture analysis. *IEEE Trans. Pattern analysis and Machine intelligence*, 8:118–125, 1986.